УДК 004.4

ОРГАНІЗАЦІЯ ПОВНОТЕКСТОВОГО ПОШУКУ У ВЕБ-ПРОЕКТАХ

Городецький О.О.
НТУУ «Київський Політехнічний Інститут»
кафедра автоматизованих систем обробки інформації та управління
E-mail: gor.saha@gmail.com

Анотація

Городецький О.О. Організація повнотекстового пошуку у Веб-проектах. *Розглянуто методи пошуку в системах зі складним інформаційним простором. Окремі з цих методів реалізовано в рамках пошукової системи Веб-проекту. Встановлено та описано основні принципи побудови пошукової системи, етапи її розробки з обґрунтуванням архітектурних рішень з точки зору особливостей проекту. Визначено показники продуктивності пошукової системи та перспективи масштабованості.*

Загальна постановка проблеми. Будь-яка система зі складним інформаційним простором повинна володіти технічними засобами для пошуку інформації та покращення контролю над пошуковим процесом. Побудова цих засобів є задачею наукових напрямків інформаційного пошуку (information retrieval), покращення взаємодії людини та комп'ютера (human-computer interaction), які стали основою для нового напрямку – інформаційний пошук в процесі взаємодії людини та комп'ютера (Human–computer information retrieval, далі HCIR)**.** Метою HCIR є подолання розриву між даними (raw data) та знаннями (оброблені дані або інформація, яка надає контекст, необхідний для наступної пошукової ітерації) [2]. Особливо це відноситься до неструктурованої документальної текстової інформації у локальних або в гіпертекстових базах даних (Інтернет).

Методи HCIR

Методи HCIR мають на меті покращити репрезентацію пошукових результатів, що сприяє вибору потрібних користувачу документів або подальшому уточненню запитів. До таких методів належать:

- автоматичне перефразування запитів;
- автоматичне доповнення слів;
- виправлення орфографічних помилок;
- фасетний пошук (класифікація інформації по певним аспектам [1]);
- визначення релевантності;
- сортування по додатковим або мета даним документів;
- пошук схожих документів [2].

Мета дослідження. Об'єктом даного дослідження є соціальний сайт новин, користувачі якого мають змогу додавати новини та коментарі. Предметом дослідження є пошукові засоби сайту, які слід спроектувати та розробити з врахуванням особливостей предметної області. В процесі проектування обґрунтовано архітектурні рішення та основні принципи побудови пошукової системи сайту.

Вимоги до пошукової системи сайту описуються діаграмою прецедентів (рис. 1).

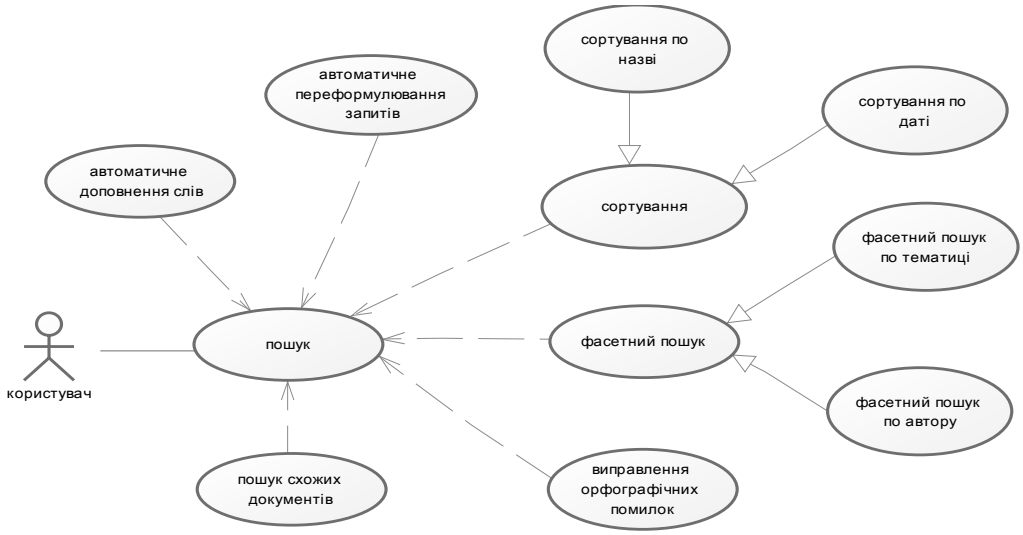

Рисунок 1 – Функціональні можливості пошукової системи

Архітектура проекту. Концептуальна схема сутностей предметної області представлена на ER-діаграмі (рис. 2)

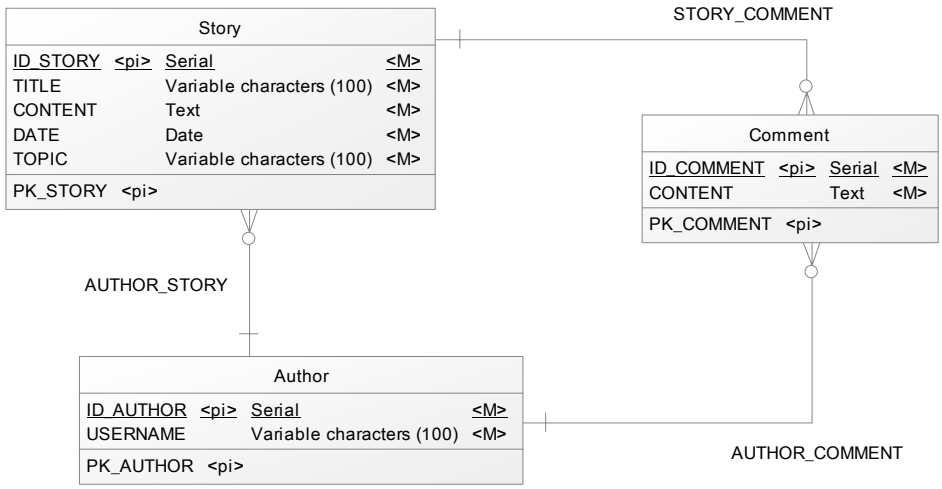

Рисунок 2 –. ER-діаграма

Соціальний сайт новин (проект Social News) реалізовано засобами Django framework, база даних – PostgreSQL. В якості пошукової системи обрано Solr на платформі Apache Lucene. Solr – самостійний сервер, який надає широкі пошукові можливості в якості веб-сервіса. Solr обмінюється повідомленнями по протоколу HTML в форматі XML або JSON. Solr підтримує кластеризацію та реплікацію на декілька серверів, зберігання додаткових полів документів, фасетний пошук, фільтрацію та сортування [6].

Для зв'язку Social News з Solr використано Haystack application, що фактично є пошуковим фреймворком для Django, та окрім Solr API реалізовує програмні інтерфейси для деяких інших пошукових систем – Elasticsearch, Whoosh, Xapian [3]. Архітектура проекту представлена на рисунку 3.

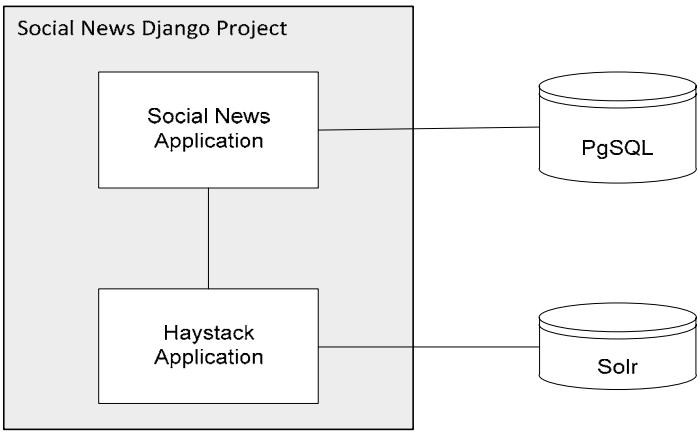

Рисунок 3 – Архітектура проекту Social News

Основні концепції проектування пошукових засобів

Документи текстових даних формуються за допомогою наперед визначених шаблонів, в які підставляються значення відповідних полів для кожної сутності з таблиці бази даних. В окремих випадках, якщо об'єкт зв'язаний з іншими, є сенс додавати в шаблон поля зв'язаних об'єктів.

Для подальшої обробки результатів запитів шляхом фільтрації або сортування з документами асоціюють додаткові атрибути. Зазвичай це числові атрибути, дати або денормалізовані дані зв'язаних об'єктів.

Функція автоматичного доповнення слів вимагає використання моделей N-грам [4]. В бібліотеці Haystack їм відповідають поля NgramField та EdgeNgramField. Останнє розбиває послідовність символів по пробілам і використовується для більшості мов.

Сервер Solr зчитує конфігураційний файл schema.xml, де описана схема індексу з вищеописаними атрибутами. Налаштування Solr визначаються в конфігураційному файлі solrconfig.xml, де задаються обробники запитів, підключені компоненти, параметри реплікації на інше [5].

Реалізація пошукової системи засобами Haystack та Solr

Вибір полів таблиць, які будуть додані в текстовий індекс. Шаблон документів включає атрибути Story.title, Story.content, Comment.content усіх коментарів, зв'язаних з об'єктом Story, та Author.username автора новини.

Визначення додаткових атрибутів індексів. Для групування документів по атрибутам теми та автора новини відповідно додано поля topic та author; для сортування – поля date та title. Атрибут autocomplete типу EdgeNgramField додано для реалізації автодоповнення слів. За допомогою поля spellsuggest типу FacetCharField реалізовано функцію виправлення помилок слів.

Клас StoryIndex на мові Python описує структуру індексу та його зв'язок з полями об'єктів класу Story.

```
class StoryIndex(indexes.SearchIndex, indexes.Indexable):
    # index fields define index' structure
    text = indexes.CharField(document=True, use_template=True)
    title = indexes.CharField(model_attr="title")
    date = indexes.DateTimeField(model_attr='date')
    topic = indexes.CharField(model_attr="topic", faceted=True)
    author = indexes.CharField(faceted=True)
    autocomplete = indexes.EdgeNgramField()
```

```
spellsuggest = indexes.FacetCharField()
# index relation to data model
def get_model(self):
    return Story
# rules that define index' relation to data model
def prepare(self, obj):
    prepared_data = super(StoryIndex, self).prepare(obj)
    prepared_data['author'] = obj.author.username
    prepared_data['spellsuggest'] = prepared_data['text']
    prepared_data['autocomplete'] = prepared_data['text']
    return prepared_data
```

StoryIndex			
STORY_INDEX_ID	<pi>	Serial	<M>
TEXT		Text	<M>
TITLE		Variable characters (100)	
DATE		Date	
TOPIC		Variable characters (100)	
AUTHOR		Variable characters (100)	
AUTOCOMPLETE		Variable characters (256)	
SPELLSUGGEST		Variable characters (256)	
PK_STORY <pi>			

Рисунок 4 – Структура індексу

У файлі solrconfig.xml визначено підключення додаткових компонент: FacetComponent, MoreLikeThisComponent, SpellCheckComponent.

Визначення стратегії оновлення індексів. Існує декілька підходів до оновлення індексів. Вибір одного з них залежить від таких факторів: частота записів в базу даних та необхідність постійно оновлених індексів. Перший підхід – індексація в реальному часі, - індекс оновлюється кожного разу, коли відбувається запис або видалення з бази даних. Це може бути нераціонально, якщо частота записів в базу даних велика, хоча перевагою є завжди оновлений індекс. Інший підхід полягає в тому, щоб оновлювати індекс за наперед визначеним розкладом, що можна реалізувати за допомогою планувальника задач cron в UNIX-подібних системах [3].

Визначення продуктивності пошукової системи.

Для впровадження та підтримки експлуатації пошукової системи важливо знати деякі її параметри. Найбільш важливі з них такі:

- швидкість індексації (наскільки швидко пошуковий сервер обробляє документи та заносить їх у свій індекс; зазвичай вимірюється в Мб тексту за секунду);
- швидкість переіндексації (в процесі роботи документи змінюються та додаються нові, якщо сервер підтримує інкрементну індексацію, то обробляються тільки нові документи, а оновлення всього індексу відкладається на більш сприятливий для цього час);
- швидкість пошуку та розмір бази даних (взаємопов'язані параметри – велика кількість документів вимагає більшого часу на пошук; як правило час пошуку стає критично великим, коли кількість документів досягає порядку мільйонів);
- розмір індексу (залежить від розміру бази даних та може бути важливим для оцінки параметрів серверу, де розгорнута пошукова система) [6].

Далі приведені тестові результати побудованої системи, які показали, що при 10 тисячах проіндексованих документів сервер Solr здатний обробляти близько 90 запитів за секунду. Тести проводились на комп'ютері з процесором Intel Core i3-380M 2.53 ГГц, RAM DDR3 1066 МГц 3072 Мб.

Таблиця 1 – Показники продуктивності пошукової системи

Кількість документів	Розмір бази даних, Мб	Розмір індексу, Мб	Час індексації, мс	Час пошуку, мс
1000	2,35	25,5	18986	10,79
5000	12,53	110,9	96361	10,91
10000	25,92	229,4	199365	11,07

Висновки

В роботі досліджені методи інформаційного пошуку, які пристосовані для покращення процесу взаємодії людини та комп'ютера. Правильна репрезентація пошукових результатів сприяє вибору потрібних користувачу документів або подальшому уточненню запитів. Для кожної інформаційної системи важливо визначити ці методи на етапі проектування, оскільки від них залежать важливі архітектурні рішення цілої системи. Також значну роль відіграє інформація, яка підлягає пошуку.

Найпоширенішими методами наукового напрямку інформаційного пошуку в процесі взаємодії людини та комп'ютера є автоматичне перефразування запитів, доповнення слів, фасетний пошук, пошук схожих документів, які було реалізовано для соціального сайту новин. Позначено основні кроки етапів проектування та розробки пошукової системи сайту.

В якості пошукової платформи обрано Apache Solr, яка є досить гнучкою та надає широкі можливості налаштування, підключення плагінів та компонент. Solr оптимізований під Веб системи з великим навантаженням, підтримує масштабування — реплікацію та шардинг в складі платформи, що може стати особливо актуальним зі зростанням кількості користувачів сайту.

Визначено показники продуктивності розробленої пошукової системи. При 10 тисячах проіндексованих документів сервер Solr здатний обробляти близько 90 запитів за секунду, що є хорошим результатом. Перспектива збільшення обсягу бази даних вимагає масштабування системи, в якої є потенціал як до вертикального масштабування (покращення продуктивності серверу), так і до горизонтального (реплікація та шардинг засобами Apache Solr).

Список літератури

1. Faceted search [Electronic resourse] / Інтернет-ресурс. - Режим доступу: http://en.wikipedia.org/wiki/Faceted_search. - Загол. з екрану.

2. Human–computer information retrieval [Electronic resourse] / Інтернет-ресурс. - Режим доступу: http://en.wikipedia.org/wiki/Human%E2%80%93computer_information_retrieval. - Загол. з екрану.

3. Haystack 2.0.0-beta documentation [Electronic resourse] / Інтернет-ресурс. - Режим доступу: http://django-haystack.readthedocs.org/en/latest/index.html. - Загол. з екрану.

4. N-gram [Electronic resourse] / Інтернет-ресурс. - Режим доступу: http://en.wikipedia.org/wiki/N-gram. - Загол. з екрану.

5. Новые возможности Apache Solr [Интернет-ресурс]. - Режим доступу: www/ URL: http://www.ibm.com/developerworks/ru/library/j-solr-update/. - Загол. з екрану.